

Digitalna integrirana vezja in sistemi

Andrej Trost

Vaja 1: Osnovne aritmetične operacije in povprečevalno sito

Orodje: Vivado 2025.1

Naloga: opiši in analiziraj vezje v SystemVerilogu, naredi povprečevalno sito, analiziraj izvedbe in preveri delovanje s simulacijo

Tehnologija

AMD (Xilinx) Zynq

- FPGA + ARM Cortex-A9

	Z-7007S	Z-7010	Z-7020
LUT (DFF)	14k (29k)	17k (35k)	53k (106k)
BRAM	200 KB	240 KB	560 KB
DSP	66	80	220
I/O	50-100	100	200

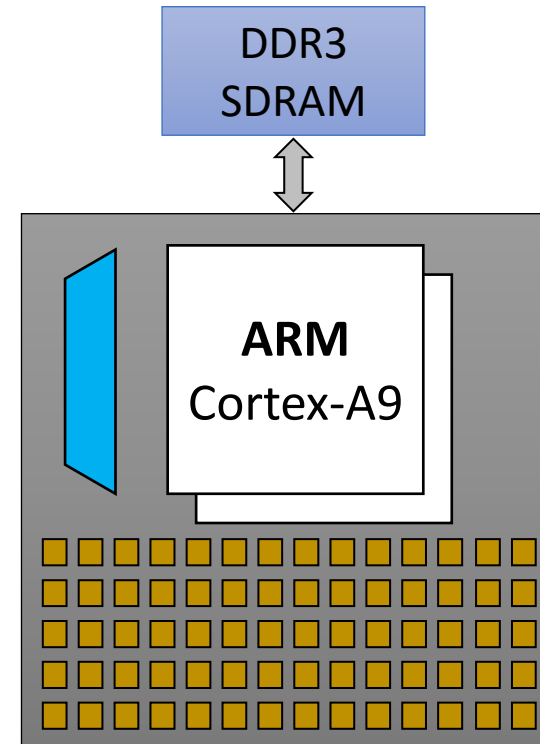
MiniZed
Cora Z7



RedPitaya



ZedBoard



FPGA: obdelava signalov
CPE: krmiljenje, povezava

- ▼ PROJECT MANAGER
 - ⚙ Settings
 - Add Sources
 - Language Templates
- ⚙ IP Catalog
- > IP INTEGRATOR
- > SIMULATION
- > RTL ANALYSIS
- ▼ SYNTHESIS
 - ▶ Run Synthesis
 - > Open Synthesized Design
- ▼ IMPLEMENTATION
 - ▶ Run Implementation
 - > Open Implemented Design
- ▼ PROGRAM AND DEBUG
 - ⬇ Generate Bitstream
 - > Open Hardware Manager

Vivado 2025

kratka pot, ne uporabljaj
presledkov in šumnikov



1. naredi mapo: c:/proj/ime
2. zaženi **Vivado**, izberi **Create Project >**
3. New Project, vnesi **Project name** in **Project location**
4. Project Type: **RTL**, Next, Next, **Default Part**: poišči xc7z010clg400 ali | Boards, Red Pitaya
5. Finish

Default Part
Choose a default AMD part or board for your project.

Parts | Boards

[Reset All Filters](#)

Category: Package: Temperature:

Family: Speed: Static power:

Search: (3 matches)

Part	I/O Pin Count	Available IOBs	LUT Elements	FlipFlops	Block RAMs
xc7z010clg400-3	400	100	17600	35200	60
xc7z010clg400-2	400	100	17600	35200	60
xc7z010clg400-1	400	100	17600	35200	60

Vivado 2025.1

aktivno okno

status

opravila

zapri

Sources & properties

konzola

poročila

The screenshot displays the Vivado 2025.1 IDE interface. The top menu bar includes File, Edit, Flow, Tools, Reports, Window, Layout, View, and Help. The main workspace is divided into several panes:

- Flow Navigator:** Contains the PROJECT MANAGER, IP INTEGRATOR, SIMULATION, RTL ANALYSIS, and SYNTHESIS sections.
- Sources:** Shows the design sources, including regop (regop.sv), constraints (constrs_1), and simulation sources.
- Synthesis Run Properties:** Displays the synth_1 run properties, including the Name, Part, and various tabs (General, Properties, Options, Log, Reports, Messages).
- Schematic:** Shows the RTL schematic diagram with registers (reg_a_reg, reg_b_reg, reg_c_reg) and a divider (RTL_DIV).
- Design Runs:** A table showing the status of the synthesis and implementation runs.

Name	Constraints	Status	WNS	TNS	WHS	THS	WBSS	TPWS	Total Power	Failed Routes	Methodology	RQA Score	LUT	QoR Suggest
✓ synth_1	constrs_1	synth_design Complete!											209	
✓ impl_1	constrs_1	route_design Complete!	62.818	0.000	0.358	0.000		0.000	0.093	0	42 Warn		209	

Dodaj opis vezja

1. PROJECT MANAGER, **Add Sources**, Add or create design sources
2. klikni **Create File**, type: **SystemVerilog**, name: logic_gates, Finish
3. definiraj priključke
4. OK
5. odpri datoteko (**Design Sources**)
klikni >, dvoklik odpre datoteko



Define Module

Define a module and specify I/O Ports to add to your source file.
For each port specified:
MSB and LSB values will be ignored unless its Bus column is checked.
Ports with blank names will not be written.

Module Definition

Module name: logic_gates

I/O Port Definitions

Port Name	Direction	Bus	MSB	LSB
a, b	input	☐	0	0
y	output	☒	1	0

OK Cancel

Vaja: operacije z registri

- nastavi omejitve v Vivadu
 - omejitve za uro, **Add Sources**, Add or create constraints
 - nastavljena perioda: 10 ns

```
create_clock -name clk -period 10.000 [get_ports clk];
```

- opiši 14-bitni seštevalnik z registri
 - preveri s simulacijo
- izvedi sintezo vezja
 - preglej zasedenost in zakasnitve (Data Path Delay)
Tcl: `report_timing -delay_type max -max_paths 1`
 - spremeni v množilnik, delilnik in ponovi korake

```
module regop(  
    input logic      clk,  
    input logic [13:0] a,b,  
    output logic [13:0] reg_c  
);  
  
    logic [13:0] reg_a, reg_b;  
    logic [13:0] c;  
  
    always_ff @(posedge clk) begin  
        reg_a <= a;  
        reg_b <= b;  
        reg_c <= c;  
    end  
  
    assign c = reg_a + reg_b;  
endmodule
```

vezje	LUT / FF / DSP	zakasnitev [ns]	log. nivojev
add 14			
mul 14			
div 14			

Naloga: povprečevalno sito I

- povprečje zadnjih treh vzorcev
- notranji registri
- preveri delovanje s simulacijo
- preizkusi opcije za deljenje (1/3)
 - a. približek s potenco števila 2
 - b. celoštevilsko deljenje
 - c. približek z množenjem, 11/32

```
module avg3(  
    input logic clk,  
    input logic signed [13:0] x,  
    output logic signed [13:0] avg  
);  
    logic signed [13:0] z [3];  
    logic signed [15:0] sum;  
  
    always_ff @(posedge clk) begin  
        z[0] <= x;  
        z[1] <= z[0];  
        z[2] <= z[1];  
        avg <= ...  
    end  
  
    assign sum = z[0] + z[1] ...
```

vezje	LUT / FF / DSP	zakasnitev [ns]	log. nivojev
a.			
b.			
c.			

Povprečevalno sito II

- opiši sito z N-odcepi
 - N naj bo parameter vezja
- deklariraj zbirko N 14-bitnih vektorjev
- uporabi zanko za pomikalni register in novo zanko za izračun vsote
 - Koliko bitov potrebuješ za vsoto?
- naredi sintezo vezij za različne vrednosti parametra: $1 < N < 100$

```
module avgN #(parameter int N = 4)(  
    input logic clk,  
    input logic signed [13:0] x,  
    output logic signed [13:0] y  
);
```

→ `logic signed [13:0] z [N];`

```
// pomikalni register  
always_ff @(posedge clk) begin  
    z[0] <= x;  
    for (int i = 1; i < N; i++) begin  
        z[i] <= z[i-1];  
    end  
end
```